**IAM Groups**

I mentioned to use managed policies for baseline permissions, permissions that you want to assign to many different identities, and then best practice is to use inline policies for any exceptions to that. So if you want to allow more access to resources or to add explicit denies to limit access to resources. That's where you use inline policies. You've also learned when to use IAM users for real users, service accounts, or applications when you can visualize the things that will use those IAM users.

**IAM groups serve a specific function in IAM. They make administration easier. An IAM group is essentially a collection of users. Users could be members of zero or more groups and groups can have many members.**

So in this particular example, I have got a fictitious company called Feline Incorporated and Feline Incorporated have an AWS account. Now, the bosses of Feline Incorporated are two cats named Ruffle and Butters, and they're both members of a group called Admins. Now, in addition to the two cats, the company has a number of human slaves. We've got a feeders group, and this group is responsible for maintaining the quantity of food in the bowls as well as providing some yummy treats. The members of that group are Ashley, Adrian, and Mark. We've also got a strokers group and Ashley is a member of that group as well, an IAM user could be a member of multiple IAM groups and an IAM group can have multiple IAM users, so keep that in mind. Now, you've learned earlier in the course that IAM policies could be directly associated with IAM users, either as inline policies or by associating manage policies with IAM users. We can also do the same with groups. So a group can either have an inline policy or a group could be associated with the managed policy and how that works is that any members of that group's in this case, Ruffles and Butters. They'll both receive this IAM policy, which is the admin IAM policy for the account. So if you're a member of a group, you automatically get all of the policies that are either inline on that group or attached to that group. They flow down onto your IAM users. So when AWS is evaluating which policies apply, it looks at the policies that are on your IAM user, either inline or managed policies, as well as looking at the inline or manage policies that are associated with any groups that your user is a member of. Essentially, IAM groups serve an administrative function because what you can do is group up your IAM users and then apply policies on the group level. That means you don't have to worry about individuals. You can have an admin group, maybe a feeders group, a strokers group, things of a more serious nature, like finance and operations and IT. All of those groups can exist in your business, and their groups can be assigned specific IAM policies which any members of that group gain access to. Now remember, AWS evaluate all of the policies in one big bunch and so if you do have any policies which flow in from an admin group, for example, any of the policies for the feeders group or the strokers group. If you're a member of multiple groups, you get multiple policies AWS evaluate it in the same way **they look for any explicit denies and if there's an explicit deny that always overrides. Next they look for any explicit allows if there's an explicit allow then you're allowed, and if there isn't an applicable policy well, in that case, you're implicitly denied and an implicit deny is essentially, this default deny, which you get if you don't have any explicit allows.** So adding groups into the mix doesn't really alter anything. It just means that if you're a member of one or more groups, then you get one or more sets of policies.

So let's just have a look at that. So if go to IAM there isn't a great deal that you need to worry about from a group perspective. So at the moment I have two IAM users in this account test users and my personal use a.cantrill. I'm going to go to groups, I'm going to create a group, and I'm going to call it feeders. So I'll call the group feeders. I'll click on Next step. I won't apply anything at this point. I'll just go ahead and create the group. Now I'll open the group and I'll point out a number of things first. **This group does have an ARN but groups are really interesting in the you can actually referenced them from other policies.** So what do I mean by this? Well, you haven't actually learned about resource policies yet, but **a resource policy allows you to set a policy on a particular resource** let's say an S3 bucket and then on that policy can say, I want this user, this user, and this user to have access. What you **can't do is point at a group from a resource policy**. You can't say on a resource policy I want this group to have access. You can only say I want identities so users and roles. So while you can have policies, identity policies associated with groups, what that's actually doing is it's not really applying them to the group. It's applying them to the group on the proviso that that will flow on to specific identity so IAM users. It's the identities in the groups that end up getting those policies. **Groups aren't really a real thing in AWS. They're just an admin construct. They're an admin entity you can add users to, and it helps you organize them, and you can apply policies which will flow on to users but they're not really in their own right, and that's important to know.** Obviously, with a group, you can apply permissions. I mentioned how you can do inline policies so you can apply inline policies and attach those the group attach managed policies and then any policies that are associated with this group passed down to any members. They allow you to group users and then to manage things based on those groups.

Exam hints:

* **Groups are an admin only feature, so they allow you to group IAM users.**
* **Groups can contain many IAM users, and users could be in many groups.**
* **IAM inline policies can be added to groups, and these flow on to the IAM users who are members.**
* **Managed IAM policies can be attached to IAM groups, and they also flow on to IAM users who are members.**
* **Groups are not true identities. They cannot be referenced from resource policies.** You could not take a resource policy on an S3 bucket and with this example say that you wanted to give the feeders group access to the bucket that is not possible. You'd have to reference each of the IAM users individually, so you cannot reference groups from resource policies and additionally,
* **Groups have no credentials**. If you see any exam questions that talk about logging in with groups or accessing things with groups that's not how it works. Groups have no credentials. You can't log in to them. They have no method of authenticating to IAM. Their sole function is to allow policies to be attached to them, allow users to be members of them, and allow those policies to flow on to the individual IAM users.